METHOD AND SYSTEM FOR REMOTE MEASUREMENT
OF NETWORK TRAFFIC

BACKGROUND OF THE INVENTION

5   The present invention relates generally to communication networks, and more particularly, methods and systems for remote measurement of traffic in a network, such as the Internet.

Existing methods and systems for measuring traffic through the Internet are quite intrusive. For example, some systems require at the very least installation of additional 10   software at each and every router and web server in the Internet. Others require the installation of traffic monitoring hardware and software at segments of the Internet. As the Internet continues to expand worldwide, however, these existing methods and systems have become prohibitively expensive and impractical. It would be expensive and impractical, if not impossible, to install traffic monitoring devices in each and every router 15   and web server worldwide to measure the total Internet traffic between or within countries.

DESCRIPTION OF THE INVENTION

To overcome the above and other disadvantages of the prior art, methods and systems are provided for determining inter-region and intra-region traffic in one or more networks. A region may be any geographical area including, for example, a town, city, 20   province, state, country, and/or continent.

To determine the network traffic flowing between a first region and a second region, one or more routers in the first and second regions are identified along with their respective links connecting the routers in the first region to those in second region. A processor located remotely from the routers in the first and second regions then determines 25   the used bandwidth of the identified links over a period of time. Based on the determined used bandwidth, the processor determines the network traffic flowing between the first and second regions.

To determine the network traffic within a region, a sample of hosts in the region are selected. The processor identifies the last pairs of identifiable routers on respective 30   routes from the processor to the sample hosts and the respective links connecting the identified last pairs of routers. The processor then determines the used bandwidth of the identified respective links. Based on the determined used bandwidth, the processor determines the network traffic within the region.

The description of the invention and the following description for carrying out the best mode of the invention should not restrict the scope of the claimed invention. Both provide examples and explanations to enable others to practice the invention. The accompanying drawings, which form part of the description for carrying out the best mode

5   of the invention, show several embodiments of the invention, and together with the description, explain the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

In the Figures:

Figure 1 is a block diagram of a network whose total inter-region and intra-

10   region network traffic are determined, in accordance with methods and systems consistent with the present invention;

Figure 2 is a block diagram of a traffic monitor, in accordance with methods and systems consistent with the present invention;

Figure 3 is a flow chart of the steps for remotely measuring network traffic

15   between two regions, in accordance with methods and systems consistent with the present invention;

Figure 4A and 4B illustrate graphical representations of the used bandwidth or traffic intensities in a link over a period of time, in accordance with methods and systems consistent with the present invention;

20   Figure 5 is a block diagram of traffic monitor remotely measuring network traffic within a region, in accordance with methods and systems consistent with the present invention; and

Figure 6 is a flow chart of the steps for remotely measuring network traffic within a region, in accordance with methods and systems consistent with the present

25   invention.

BEST MODE FOR CARRYING OUT THE INVENTION

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the

30   same or like parts.

-2-

In accordance with an embodiment of the invention, the total inter-region traffic in a network is determined using a remote processor. To determine the network traffic flowing between a first region and a second region, the processor identifies the routers in the first and second regions along with their respective backbone links connecting the routers in the first region to those in the second region. The processor then remotely determines discrete used bandwidth samples of the identified links over a desired period of time. Based on the used bandwidth samples, the processor determines the total network traffic flowing between the first and second regions.

In another embodiment, the processor determines the total intra-region traffic in a network based on a sample of selected hosts each including one or more web servers. The processor identifies the last pairs of identifiable routers on respective routes from the processor to the sample hosts and the respective links connecting the identified last pairs of routers. The processor then remotely determines discrete used bandwidth samples of the identified respective links. Based on the used bandwidth samples, the processor determines the total network traffic within the region.

Figure 1 is a block diagram of a network 100, in accordance with methods and systems consistent with the present invention. Network 100 may include a shared network such as the Internet, which spans a plurality of regions 1, 2, and 3. A region may be any geographical area including, for example, a town, city, province, state, country, and/or continent. A region may include a plurality of hosts (not shown) and a plurality of routers 110 connecting that region via backbone links 120 to routers 110 in one or more other regions.

A host may include one or more web servers. A router may include a device that forwards packets in network 100 based on network layer and routing tables, which may be constructed by routing protocols.

Network 100 may also include a traffic monitor 130 located in region 3 and remotely from routers 110 and their respective backbone links 120 in regions 1 and 2. Alternatively, traffic monitor 130 may instead be placed anywhere in network 100, including in regions 1 and 2 so long as it is placed remotely from routers 110 and backbones 120. Traffic monitor 130 may include any processor, such a computer or a

mobile communication device capable of sending and receiving information about network 100.

Figure 2 is a block diagram of traffic monitor 130, in accordance with methods and systems consistent with the present invention. As shown, traffic monitor 130 may include a processor 200, which connects via bus 210 to a memory 220, a secondary storage 230, a network interface module 240, and an input/output module 250.

Memory 220 may include a traffic measuring program 260, an operating system 270, a region table 280, a router topology table 285, and a host table 290. As described below in detail, traffic measuring program 260 may include software, which processor 200 executes to measure total inter-region and/or intra-region traffic in network 100.

Region table 280 may include region information, such as longitude and latitude of various areas in regions 1, 2, and 3. An area may include, for example, a state, province, city, town, etc. within a region. Each entry may include the name of an area, such as a city or state and its associated longitude and latitude.

Router topology table 285 may include information about the topology of routers in network 100, such as the interconnection of routers 110 in regions 1 and 2. Each entry in the router topology table 285 may identify one or more routers 110 to which a router 110 may connect via one or more links. For example, an entry in the router topology table 285 may identify a router 110 in region 1 connecting to a router 110 in region 2 via a backbone link 120. Each entry may include longitude, latitude, and network address information, such as an Internet Protocol (IP) address for a router 110.

A user may populate the entries in the router topology table 285 by downloading the router topology information from Cooperative Association of Internet Data Analysis (CAIDA), which may be accessed on-line at www.caida.org. The web page www.caida.org/tools/measurement/skitter/research.html describes the procedure for downloading the CAIDA database.

Host table 290 may include information about a sample of hosts in regions 1, 2, and 3. Each entry in the host table 290 may include, for example, a host name, a host address, and associated region information. A host name may include a web name, such as "www.telcordia.com." A host address may include a network address, such as an IP

address of a host in network 100. Region information may include, for example, the name of the country, state, city, town, etc. where a host is located.

A user may populate the entries in host table 290 by downloading sample host information from, for example, Netsizer, a tool developed and made available on-line by Telcordia Technologies, Inc. at www.netsizer.com. A Netsizer database includes names and addresses of sample hosts worldwide. The user may purchase from Telcordia Technologies, Inc. a copy of the database and load portions or all of the database onto host table 290. The user may purchase a copy of the Netsizer database through the web page www.netsizer.com/memebers2.html.

Secondary storage 230 may include a computer readable medium such as a disk drive and a tape drive. From the tape drive, software and data may be loaded onto the disk drive, which may then be copied onto memory 220. Similarly, software and data in memory 220 may be copied onto the disk drive, which may then be loaded onto the tape drive.

Network interface module 240 may include hardware and software for sending and receiving information from network 100.

Input/Output interface 250 may include, for example, a key board or a key pad and a display unit.

Figure 3 is a flow chart of the steps traffic measuring program 260 in traffic monitor 130 performs to remotely measure network traffic between two regions, for example regions 1 and 2, in accordance with methods and systems consistent with the present invention. Traffic measuring program 260 may search the router topology table 285 to identify those routers 110 in region 1 that connect to routers 110 in region 2 (step 300). To identify such routers, traffic measuring program 260 may cross-reference the longitude and latitude information in the entries of the router topology table 285 against the region table 280. For each entry in the router topology table 285, if the longitude and latitude of a router 110 falls within region 1, then traffic measuring program 260 may determine whether the longitude and latitude of any other router 110 connected to that router falls within region 2. If the longitude and latitude of a connected router 110 falls within region 2, traffic measuring program 260 may select that entry. Otherwise, traffic monitoring program 260 may retrieve the next entry in the router

topology table and repeat the above steps until all routers 110 in region 1 that connect to routers 110 in region 2 are identified.

Traffic measuring program 260 may then identify the backbone links connecting the routers according to the network addresses of the routers at each end of the backbone links (step 310). For example, a backbone link may be identified according to the IP address of a router 110 in region 1 at one end of the backbone link and the IP address of another router 110 in region 2 at the other end of the backbone link.

Traffic measuring program 260 may then determine the used bandwidth of each of the identified backbone links connecting the routers 110 in region 1 to those in region 2 (step 320). For example, traffic measuring program 260 may invoke a getbandwidth( ) routine to estimate the used bandwidth of each identified backbone link. The getbandwidth( ) routine may be purchased from Telcordia Technologies, Inc. as part of the Netsizer tool and bounded to the traffic measuring program 260. The getbandwidth( ) routine may be purchased through the web page www.netsizer.com/memebers2.html.

Traffic measuring program may then repeatedly invoke the getbandwidth( ) routine for each identified backbone link. Traffic measuring program 260 may invoke the getbandwidth( ) routine with the following parameters: 1) the IP address of a router 110 in region 1 at one end of a backbone link, 2) the IP address of another router 110 in region 2 at the other end of the backbone link, and 3) a number of discrete used bandwidth samples to be taken from the backbone link over a period of time $P = [t_0, t_1]$. Traffic measuring program 260 then may determine a plurality of discrete used bandwidth samples for each identified backbone link 120 by repeatedly invoking the getbandwidth( ) routine for each identified backbone link 120. As an illustration, Figure 4A depicts a graphical representation of the discrete used bandwidth or traffic intensity samples for an identified backbone link 120, over the period of time $P = [t_0, t_1]$.

Based on the discrete used bandwidth samples, traffic measuring program 260 may then determine the total network traffic flowing between regions 1 and 2 (step 330). For example, traffic measuring program 260 may first determine the mean traffic $\hat{\mu}_j$ of the identified backbone links as follows:

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} I_{ij}}{n}$$

where I_{ij} represents the used bandwidth or traffic intensity sample i of an identified backbone link 120_j, and n represents the total number of used bandwidth samples taken from the identified backbone link 120_j during the period $P = [t_0, t_1]$.

5   Traffic measuring program 260 may then determine the total network traffic T flowing between regions 1 and 2 over the period of time $P = [t_0, t_1]$ as follows:

$$T = P. \sum_{j=1}^{N} \hat{\mu}_j$$

where N represents the total number of backbone links 120 connecting routers 110 in
10   region 1 to routers 110 in region 2. Traffic measuring program 260 may then determine the variance V of the total traffic T as $s^2 = V(T)$, determining, for example, a 95% confidence interval for the total traffic represented as $T \pm 1.96s$.

Alternatively, after determining the discrete samples of used bandwidth for each identified backbone link 120 over a period of time $P = [t_0, t_1]$, traffic measuring
15   program 260 may fit the discrete samples to a traffic model, such as a Fourier series or a polynomial model of the form $I(t) = \alpha + \beta t + \gamma t^2 + \lambda t^3$, where t represents time, and α, β, γ, and λ represent the parameters of the traffic model.

Traffic measuring program 260 may then invoke any commercially available software, such as SAS, SPSS, Splus, etc. to estimate each of the parameters. These
20   software packages are developed by SAS Institute, SPSS, and Splus, respectively. Further information on SAS, SPSS, and Splus software packages is available on the following web sites: www.sas.com, www.spss.com, and www.splus.com.

Traffic measuring program 260 may use a least squares method, such as that of the Splus software to estimate the parameters α, β, γ, and λ. Alternatively, traffic
25   measuring program may invoke the Base SAS tool disclosed in web page

http://www.sas.com/products/base/index.html. Figure 4B is a graphical representation of a resulting continuous traffic model based on the estimated parameters. Traffic measuring program 260 may then determine the total traffic T flowing between regions 1 and 2 by integrating the traffic model $I(t)$ over the period of time $P = [t_0, t_1]$, as shown by the shaded

5   area in Figure 4B.

In another embodiment, traffic measuring program 260 may select only a random sample of routers 110 in region 1 connected via backbone links 120 to routers 110 in region 2. Traffic measuring program 260 may then perform the steps 300-330 as described above to determine a plurality of discrete used bandwidth samples for each sample

10   backbone link 120. To determine the total traffic flowing between regions 1 and 2, however, traffic measuring program 260 may first estimate the total number $\hat{N}$ of backbone links 120 connecting routers 110 in region 1 to those in region 2. Based on the estimated total number of backbone links, traffic measuring program 260 may then determine the total network traffic T flowing between regions 1 and 2 as follows:

$$T = P \cdot \hat{N} \cdot \frac{\sum_{j=1}^{m} \hat{\mu}_j}{m} \quad ,$$

15

where $\hat{\mu}_j$ represents the mean used bandwidth of the sample backbone links as described above, m represents the number of the sample backbone links 120, and $\hat{N}$ represents the estimated total number of backbone links 120 connecting routers 110 in region 1 to those in region 2.

20   In yet another embodiment, traffic monitor 130 may determine the total network traffic within a region, such as region 1. Figure 5 shows a block diagram of traffic monitor 130 remotely measuring network traffic within region 1, in accordance with methods and systems consistent with the present invention. As shown, traffic monitor 130, which may be remotely located in region 3, queries and receives information

25   about a plurality of hosts 510 in region 1 via one or more routers 520, 521, and 522.

Although for the sake of simplicity Figure 5 shows a single host 510, region 1 may include a plurality of hosts 510.

A host 510 may include, for example, a load balancer 530 and one or more web servers 540, and may be identified in the network by the IP address of load balancer 530.

5　　Load balancer 530 balances the load on web servers 540 by forwarding communications destined to host 510 to web servers 540 such that the communications are evenly distributed among web servers 540. Alternatively, a host may include one or more servers directly connected to one or more routers in network 100.

Figure 6 is a flow chart of the steps traffic measuring program 260 performs to

10　　remotely measure network traffic within region 1, in accordance with methods and systems consistent with the present invention. Traffic measuring program 260 may select from host table 290 a sample of hosts 510 that include one or more web severs and located in region 1 (step 600). To identify such sample hosts 510, traffic measuring program 260 may parse each host name in the entries of host table 290 to identify those hosts 510 whose

15　　host names include the term "www."

Alternatively, traffic measuring program 260 may invoke a portscan( ) routine to determine whether a host in region 1 includes a web server. The parameters of the portscan( ) routine may include the IP address of a host in region 1, which traffic measuring program 260 may retrieve from host table 290, and a port number 80. If

20　　portscan( ) returns a value indicating that port number 80 in a host is open, traffic measuring program 260 may determine that the host includes a web server and selects that host as a sample host. Otherwise, traffic measuring program 260 may determine that the host does not include a web server. Traffic measuring program 260 may then retrieve the next entry in host table 290 and repeats this step until a desired number of sample hosts are

25　　identified.

Traffic measuring program 260 may then determine the last identifiable pairs of routers on respective routes to sample hosts 510 by, for example, trace routing each identified sample host 510 (step 610). For each sample host, traffic measuring program 260 may invoke a traceroute( ) routine to identify routers on a route from traffic

30　　monitor 130 to the sample host 510. For a particular route, traffic measuring program 260 may select the last two identifiable routers in a sequence with the shortest hop to the

sample host 510. In the embodiment of Figure 5, the last two identifiable routers on a route from traffic monitor 130 to the sample host 510 are routers 521 and 522, which are connected to each other via link 550.

5   Traffic measuring program 260 may then identify the links connecting each of the selected pair of routers (step 620). For example, traffic measuring program 260 may identify each link according to the IP addresses of the pair of routers at each end of the link. Traffic measuring program 260 may determine discrete used bandwidth samples of each identified link over a period of time $P = [t_0, t_1]$ (step 630). For example, traffic measuring program 260 may estimate the used bandwidth of each identified link using the

10   getbandwidth( ) routine.

Alternatively, if a sample host 510 is pingable, traffic measuring program 260 may identify links connecting the sample host 510 to the last identifiable router with the shortest hop to the sample host 510. Traffic measuring program 260 may determine discrete used bandwidth samples of the identified link over a period of time $P = [t_0, t_1]$ by

15   invoking the getbandwidth( ) routine.

Based on the discrete samples of used bandwidth of the identified links, traffic measuring program 260 may then determine the total traffic within region 1 as follows (step 640): Traffic measuring program 260 may first determine the mean traffic $\hat{\mu}_j$ of the identified links as follows:

20

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} I_{ij}}{n}$$

where I_{ij} represents the used bandwidth sample i of an identified link j, and n represents the total number of used bandwidth samples taken from the identified link j by traffic measuring program 260 over the period of time $P = [t_0, t_1]$.

Traffic measuring program 260 may then estimate the total number $\hat{N}$ of

25   hosts 510 in region 1 by invoking, for example, a host_estimate( ) routine of the Netsizer tool. The host_estimate( ) routine may be purchased from Telcordia Technologies, Inc. from web page www.netsizer.com/members2.html.

Based on the estimated total number of hosts in region 1, traffic measuring program 260 may then determine the total traffic within region 1 as follows:

$$T = P \cdot \hat{N} \cdot \frac{\sum_{j=1}^{m} \hat{\mu}_j}{m}$$

5   where $\hat{\mu}_j$ represents the mean used bandwidth of an identified link j as described above, m represents the number of identified links j, and $\hat{N}$ represents the estimated total number of hosts in region 1, as identified by traffic measuring program 260. Traffic measuring program 260 may then determine the variance V of the total traffic T as $s^2 = V$ (T), determining, for example, a 95% confidence interval for the total traffic represented as

10    $T \pm 1.96s$.

Alternatively, after determining the discrete samples of used bandwidth of each identified link over a period of time $P = [t_0, t_1]$, traffic measuring program 260 may then fit the determined used bandwidth I to a traffic model, as described with respect to step 330 of Figure 3.

15   While it has been illustrated and described what are at present considered to be preferred embodiments and methods of the present invention, it will be understood by those skilled in the art that various changes and modifications may be made, and equivalents may be substituted for elements thereof without departing from the true scope of the invention. One skilled in the art will appreciate that all or part of the systems and

20   methods consistent with the present invention may be stored on or read from computer-readable media, such as secondary storage devices, like hard disks, floppy disks, and CD-ROM; a carrier wave received from a network such as the Internet; or other forms of ROM or RAM. This invention should be limited only by the claims and equivalents thereof.

In addition, many modifications may be made to adapt a particular element,

25   technique or implementation to the teachings of the present invention without departing from the central scope of the invention. Therefore, it is intended that this invention not be

limited to the particular embodiments and methods disclosed herein, but that the invention include all embodiments falling within the scope of the appended claims.